

- The CPU of 8051 works only on binary values (0's and 1's)
- program consists of 0's and 1's is called machine language program.
- Any Assembly program must be converted (translated) into machine language program. using program called Assembler.

Assembly

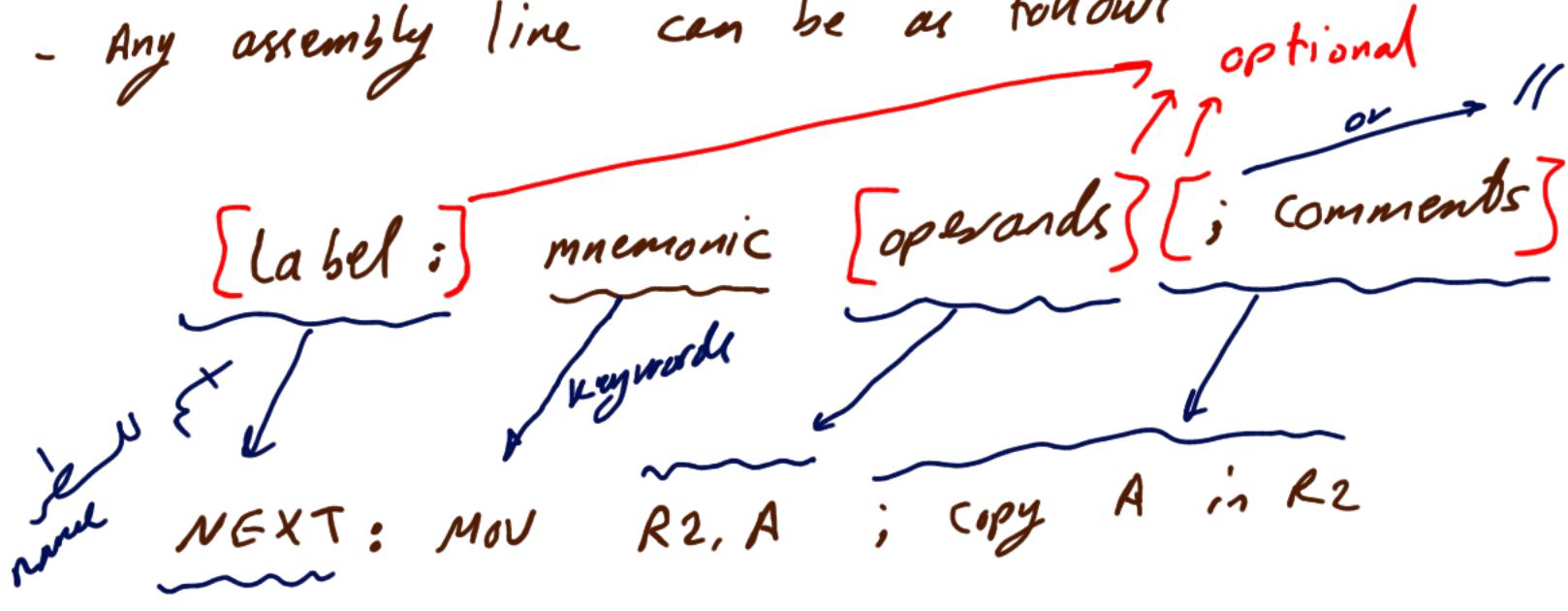
MOV A, R2
ADD A, R3
:



machine language

(2)

- Any assembly line can be as follows

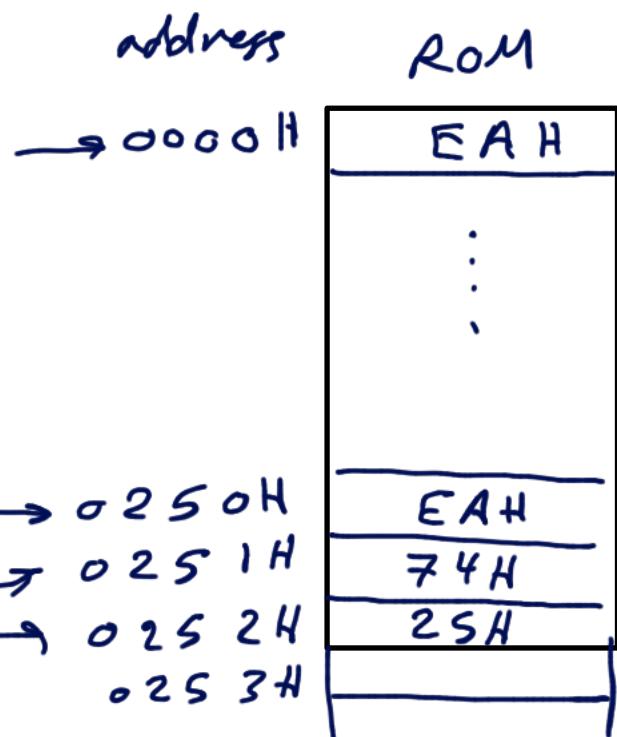


- Any assembly line can be

- Directives (pseudo instructions): Do not generate machine languages (code) but they are used by assembler itself.
- Comments don't generate any machine code.

① ORG : (origin) ^{the assembler}
 is used to indicate [↑] the beginning address of source file.

ORG 0000H
 MOV A,R2



ORG 0250H
 MOV A,R2
 MOV A,#25H
 END

② END : is used to indicate to the assembler the end
 of source file.

address	rom	
1 0000	↓	ORG 0H ;start at location 0
2 0000 7D25		MOV R5,#25H ;load 25H into R5
3 0002 7F34		MOV R7,#34H ;load 34H into R7
4 0004 7400		MOV A,#0 ;load 0 into A
5 0006 2D		ADD A,R5 ;add contents of R5 to A
6 0007 2F		;now A = A + R5 ADD A,R7 ;add contents of R7 to A
7 0008 2412		;now A = A + R7 ADD A,#12H ;add to A value 12H
8 000A 80FE HERE:		;now A = A + 12H SJMP HERE ;stay in this loop
9 000C		END ;end of asm source file

③ DB : Define byte

is used to define 8-bit (1 byte) of data in memory (ROM).

Data could be

- ASC II
- binary
- octal
- hex
- decimal

ORG 197H

DB 28H

DB 25

DB 10011100B

DB 'F'

DB '2'

DB 0E4H

DB "Hi 80SI"

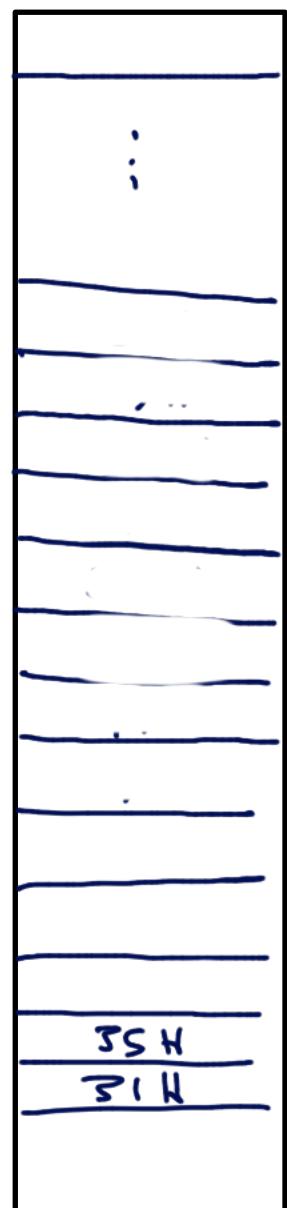
ORG 300H

DB 21H

ORG 197H

DB 28H, 25, 10011100B, 'F', ...

ROM
0000H



0300H [21H]

④ EQU : equate $\xrightarrow{\text{by 1 or more}}$
is used to define constant without occupying
a memory location in ROM.

(S)

NUM EQU 15H

:

MOV R2, #NUM

ADD A, #NUM

⑤ Bit : is similar to EQU but it is used with one bit

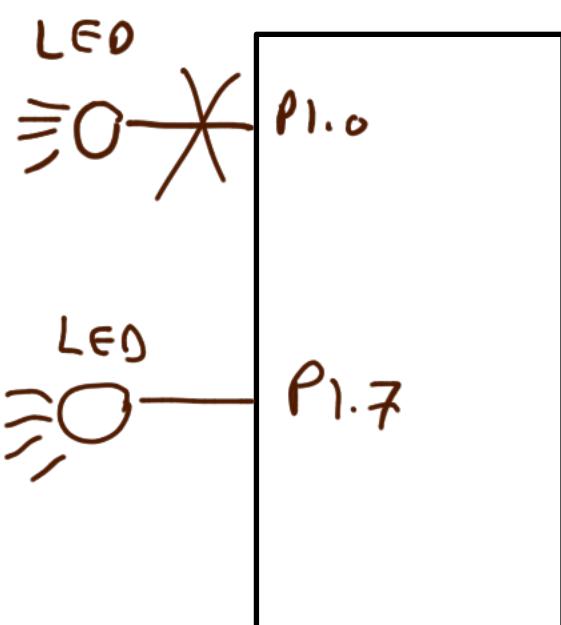
SETB P1.0

CLR P1.0

LED Bit P1.0

SETB LED

CLR LED



* Bit addressable area ($20H \rightarrow 2FH$)

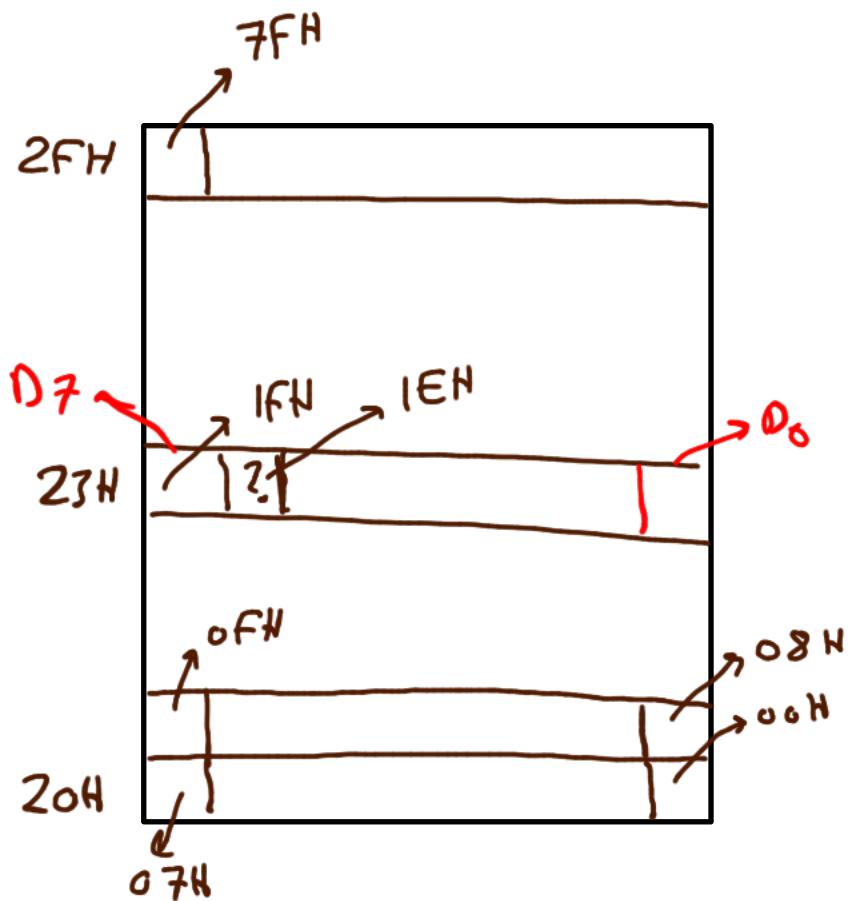
what is the byte address of
bit address $1FH$

$$1FH = 0001\ 1111B = 31$$

$$\frac{31}{8} = 3.875$$

$$[20H] + 3 = 23H$$

$$0.875 \times 8 = 7 \rightarrow \underline{\text{bit } 7}$$



what is the bit address for bit 6 in byte address $23H$

$$23H - [20H] = 3$$

$$3 \times 8 = \underline{24} D$$

$$24 \rightarrow \text{in Hex} = \underline{\underline{0001\ 1000}} = 18H$$

$$(8H + \underline{\underline{6}}) = \underline{1EH}$$

(6)

* Stack : is a section of RAM

start at bank 1 (08H)

Stack is used to store

data and addresses temporarily

* SP : stack pointer

(8-bit)

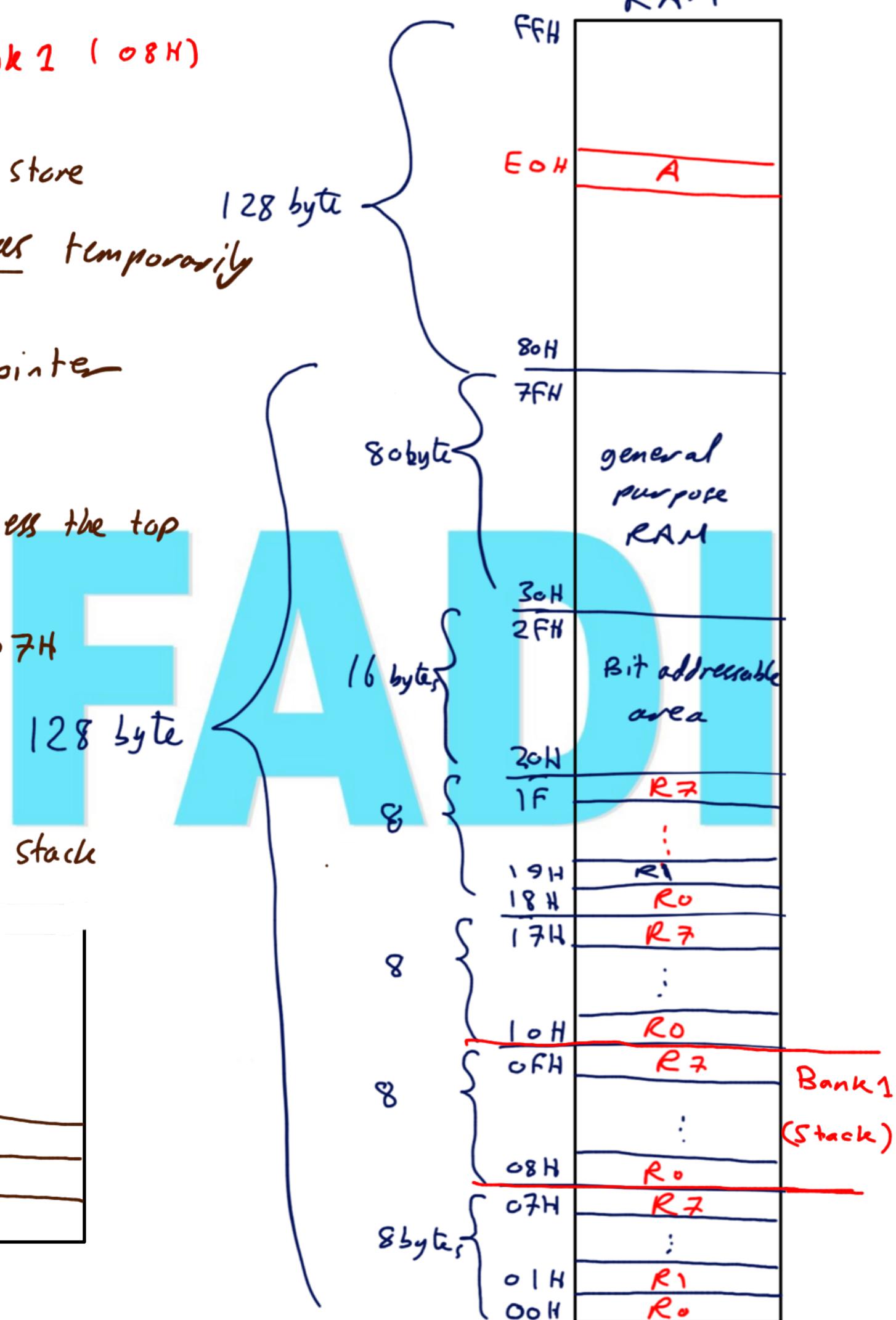
is used to access the top
of the stack

by default SP = 07H

empty stack

09H
08H

SP → 07H



stack is LIFO : Last-in - First-out

⑤

① PUSH : is used to store data in stack from Reg.
or memory location.

$$① SP = SP \underline{+ 1}$$

increment SP
by 1

$$② [SP] = mem$$

② POP : is used to remove the top of stack into
memory location

$$① mem = [SP]$$

$$② SP = SP \underline{- 1}$$

decrement SP
by 1

Show me stack and stack pointer (SP) after executing ⑨
the following code.

MOV R2, # 50H

MOV R3, # 60H

MOV R7, # 'E'

PUSH 2

PUSH 3

PUSH 7

MOV R7, # 14H

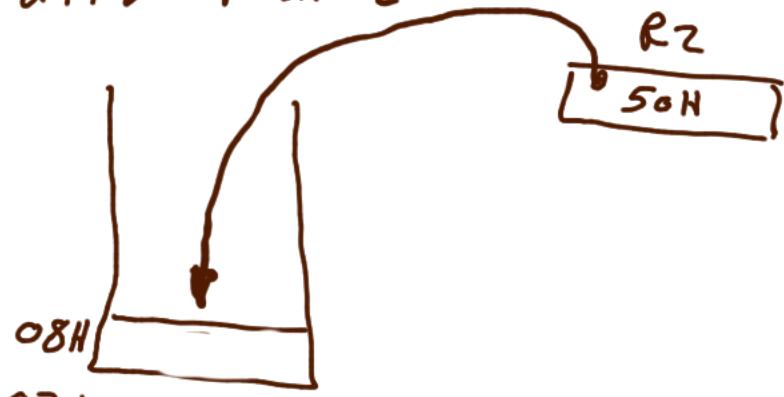
POP 7

POP 3

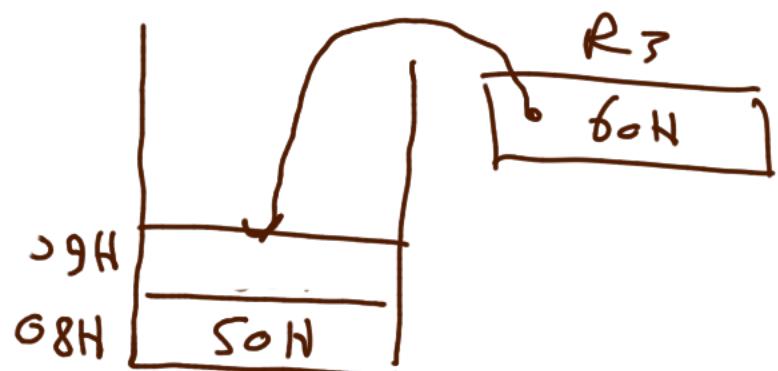
POP 2

SP → 07H

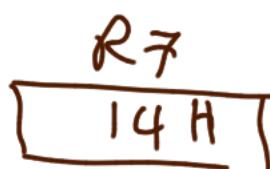
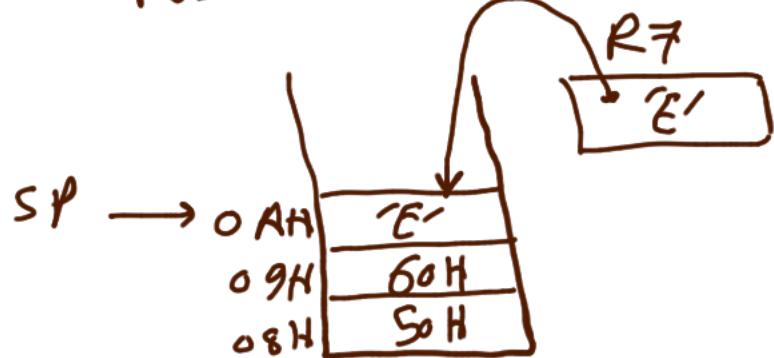
after PUSH 2



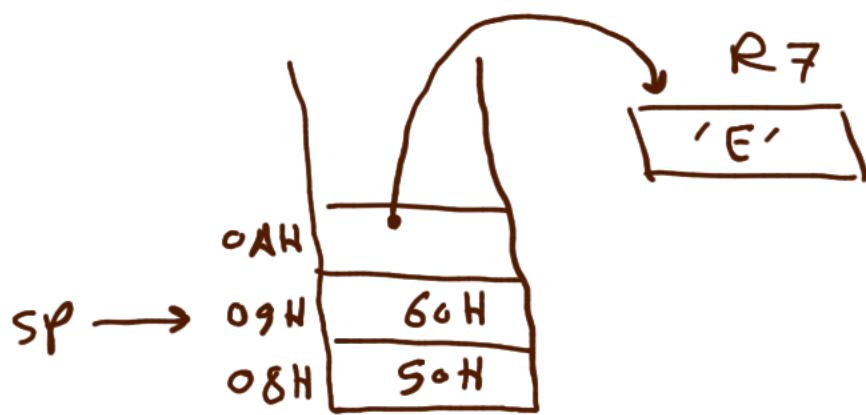
after PUSH 3



PUSH 7



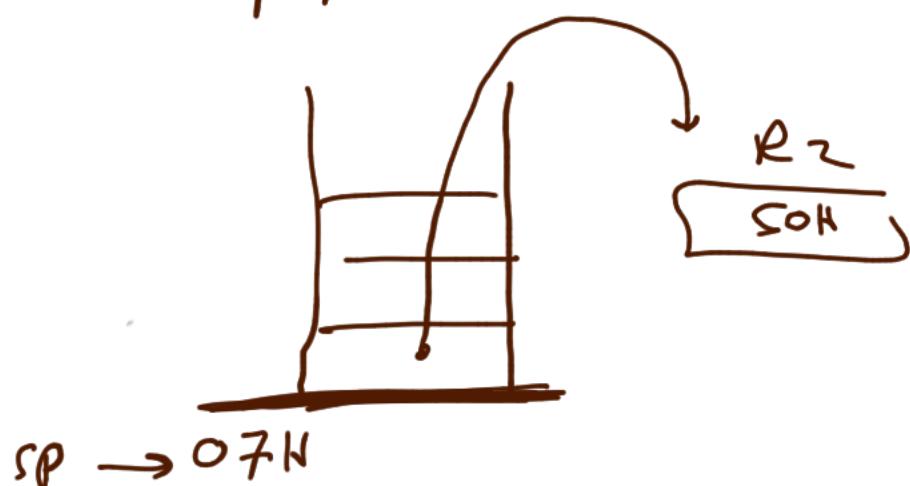
after pop 7



pop 3



pop 2



stack locations in RAM

① start at bank 1 ($08H \rightarrow 1FH$) \xrightarrow{banks}

② scratch pad RAM ($30H \rightarrow 7FH$)

using instruction

Mov SP, # 2FH

- Bit addressable area ($20H \rightarrow 2FH$) is not allowed for stack.

(11)

MOV R4, # 28H ←
MOV SP, # 59H ←
PUSH 4

